This listing of claims will replace prior versions, and listings of claims in the application.

1.   (Currently amended)   A method for a first process running on a computing device to communicate with a second process, the method comprising:

creating a process table on the computing device;

rendering the process table accessible to the first process;

associating a Universally Unique Identifier (UUID) with the second process;

creating an entry for the second process in the process table;

associating the UUID of the second process with the process entry for the second process in the process table;

configuring the second process to respond to a global event by releasing resources, reporting status, and performing a controlled shutdown;

specifying a communications task to perform; and

using the UUID of the second process to specify that the communications task be performed with respect to the second process.

2.   (Original)   The method of claim 1 wherein creating a process table comprises creating the process table as shared memory on the computing device.

3.   (Original)   The method of claim 1 further comprising:

coordinating access to the process table and to the process entry for the second

process via software locks.

4.   (Original)   The method of claim 1 wherein specifying a

communications task to perform comprises specifying monitoring a process and

wherein the method further comprises:

writing status information about the second process into the process entry for

the second process; and

retrieving the status information about the second process by using the UUID of

the second process to access the process entry for the second process in the process

table.

5.   (Original)   The method of claim 4 wherein writing status information

comprises periodically writing a heartbeat update time and wherein the method further

comprises:

comparing the heartbeat update time in the status information to the current

time; and

determining if the second process is running based on the comparing of the

times.

6.   (Original)   The method of claim 1 wherein specifying a communications task to perform comprises specifying requesting information from a process and wherein the method further comprises:

specifying a type of information requested; and

returning the information requested to the first process.

7.   (Original)   The method of claim 6 wherein the type of information requested is selected from the set: log output, console output.

8.   (Original)   The method of claim 6 further comprising:

specifying a period of time during which to return the information requested; and

wherein returning comprises returning the information requested during the specified period of time.

9.   (Original)   The method of claim 6 wherein returning comprises returning the information requested until the first process indicates that the information need no longer be returned.

10.   (Original)   The method of claim 1 wherein specifying a

communications task to perform comprises specifying waiting for the second process to

achieve a status.

11.   (Original)   The method of claim 10 wherein the status is in the set:

initialized, debug_break, terminated.

12.   (Original)   The method of claim 1 wherein specifying a

communications task to perform comprises specifying sending a signal to the second

process.

13.   (Original)   The method of claim 12 wherein sending a signal indicates

that the process should terminate.

14.   (Original)   The method of claim 1 further comprising:

associating a UUID with a third process;

creating an entry for the third process in the process table;

associating the UUID of the third process with the process entry for the third

process in the process table;

associating the UUID of the second process with the process entry for the third

process in the process table; and

using the UUID of the second process to specify that the communications task be

performed with respect to the third process.

15. (Original) The method of claim 14 wherein the third process is a

child of the second process.

16. (Original) The method of claim 15 further comprising using the UUID

of the second process to specify that the communications task be performed with

respect to all descendents of the second process.

17. (Original) The method of claim 1 wherein the second process runs

on a second computing device distinct from the computing device on which the first

process runs.

18. (Original) The method of claim 17 further comprising:

associating an identifier of the second computing device with the process entry

for the second process in the process table;

creating a second process table on the second computing device;

creating an entry for the second process in the second process table; and

associating the UUID of the second process with the process entry for the second

process in the second process table.

19.   (Original)   The method of claim 18 wherein specifying a

communications task to perform comprises specifying monitoring a process and

wherein the method further comprises:

writing status information about the second process into the process entry for

the second process in the second process table; and

retrieving the status information about the second process by using the UUID of

the second process to access the process entry for the second process in the second

process table.

20.   (Original)   A computer-readable medium having instructions for

performing the method of claim 1.

21.   (Original)   A method for a first process running on a computing

device to communicate with a second process and with a third process, the method

comprising:

creating a process table on the computing device;

rendering the process table accessible to the first process;

creating an entry for the second process in the process table;

creating an entry for the third process in the process table;

associating a group UUID with the process entry for the second process in the

process table;

associating the group UUID with the process entry for the third process in the

process table;

specifying a communications task to perform; and

using the group UUID to specify that the communications task be performed with

respect to the second and third processes.

22.   (Original)   The method of claim 21 wherein creating a process table

comprises creating the process table as shared memory on the computing device.

23.   (Original)   The method of claim 21 further comprising:

coordinating access to the process table and to the process entries for the

second and third processes via software locks.

24.   (Original)   The method of claim 21 wherein specifying a communications task to perform comprises specifying monitoring a process and wherein the method further comprises:

writing status information about the second process into the process entry for the second process;

writing status information about the third process into the process entry for the third process; and

retrieving the status information about the second and third processes by using the group UUID to access the process entries for the second and third processes in the process table.

25.   (Original)   The method of claim 24 wherein writing status information comprises periodically writing a heartbeat update time and wherein the method further comprises:

comparing the heartbeat update times in the status information to the current time; and

determining if the second and third processes are running based on the comparing of the times.

26.   (Original)   The method of claim 21 wherein specifying a

communications task to perform comprises specifying requesting information from a

process and wherein the method further comprises:

specifying a type of information requested; and

returning the information requested to the first process.

27.   (Original)   The method of claim 26 wherein the type of information

requested is selected from the set: log output, console output.

28.   (Original)   The method of claim 26 further comprising:

specifying a period of time during which to return the information requested;

and

wherein returning comprises returning the information requested during the

specified period of time.

29.   (Original)   The method of claim 26 wherein returning comprises

returning the information requested until the first process indicates that the information

need no longer be returned.

30.   (Original)   The method of claim 21 wherein specifying a

communications task to perform comprises specifying waiting for the second process to

achieve a status and the third process to achieve the status.

31.   (Original)   The method of claim 30 wherein the status is in the set:

initialized, debug_break, terminated.

32.   (Original)   The method of claim 21 wherein specifying a

communications task to perform comprises specifying waiting for the second process or

the third process to achieve a status.

33.   (Original)   The method of claim 32 wherein the status is in the set:

initialized, debug_break, terminated.

34.   (Original)   The method of claim 21 wherein specifying a

communications task to perform comprises specifying sending a signal to the second

and the third processes.

35.   (Original)   The method of claim 34 wherein sending a signal indicates

that a process should terminate.

36. (Original) The method of claim 21 wherein the second process runs on a second computing device distinct from the computing device on which the first process runs.

37. (Original) The method of claim 36 further comprising:

associating an identifier of the second computing device with the process entry for the second process in the process table;

creating a second process table on the second computing device;

creating an entry for the second process in the second process table; and

associating the group UUID with the process entry for the second process in the second process table.

38. (Original) The method of claim 37 wherein specifying a communications task to perform comprises specifying monitoring a process and wherein the method further comprises:

writing status information about the second process into the process entry for the second process in the second process table; and

retrieving the status information about the second process by using the group UUID to access the process entry for the second process in the second process table.

39. (Original) A computer-readable medium having instructions for performing the method of claim 21.

40-65. (Cancelled)

66. (New) A computer-readable medium having instructions for performing steps comprising:

executing a first process in a first computing device;

launching a second process in a second computing device, the second process being invoked by the first process;

identifying the second process in a shared memory included in the first computing device, the second process being identified in the shared memory with an identifier unique to the second process, the unique identifier being independent from the computing device on which the second process is running;

configuring the first process and the second process to communicate based, at least in part, on the information in the shared memory;

associating a global event with the second process; and

configuring the second process to respond to the global event by releasing resources, reporting status, and performing a controlled shutdown.

67.   (New)   The computer-readable medium as recited in claim 66, further comprising configuring the second process to periodically log heartbeat entries in the shared memory.

68.   (New)   The computer-readable medium as recited in claim 67, further comprising configuring the first process to access the heartbeat entries logged by the second process in the process table.

69.   (New)   The computer-readable medium as recited in claim 66, further comprising:

associating a set of processes with an parent identifier that identifies a parent process from which the processes in the set depend; and

in response to a termination of the parent process, canceling the processes identified by the parent identifier.

70.   (New)   A computing device comprising:

means for executing local processes on the computing device;

means for launching remote processes invoked by the local processes on other remote computing devices;

means for maintaining a process table on the computing device that includes

information about the local processes and the remote processes;

means for identifying the local processes and the remote processes with

identifiers that do not distinguish the remote processes from the local processes;

means for enabling the remote processes to update the information in the

process table; and

means for the local processes to access the updated information about the

remote processes.

71.   (New) The computing device as recited in claim 70, further comprising

means for associating a global event with each remote process.

72.   (New) The computing device as recited in claim 71, wherein the global

event includes at least one of means for releasing resources, means for reporting status,

and means for performing a controlled shutdown.

73.   (New) The computing device as recited in claim 70, further comprising:

means for defining global events associated with all of the remote processes

identified in the process table; and

means for assigning meanings to the defined global events.

74.   (New)   The computing device as recited in claim 70, wherein the

information in the process table associated with each remote process includes at least

one of a parent identifier, a group identifier, a creation time, a heartbeat time, a source

device, a target device, or a process type.